Programación II

Ejercicio del Mecánico

Dpto. LSIIS. Unidad de Programación

Objetivo: El objetivo de esta práctica es la familiarización del alumno con la programación orientada a objetos en Java.

Evaluación: Esta práctica tiene un peso de un 12,5% sobre la nota final de la asignatura. La práctica se podrá realizar en grupos de dos alumnos o de forma individual. Se mantendrá el mismo enunciado de la práctica para la convocatoria extraordinaria.

Grupos: Los grupos estarán formados por dos alumnos matriculados en el mismo semestre. Los alumnos deben estar matriculados y dados de alta en el sistema de entrega (maui) antes de proceder a realizar el registro del grupo. Ninguno de los dos alumnos deberá haber realizado entrega alguna de la práctica antes de definir el grupo. Una vez creado el grupo es el mismo para todas las prácticas en grupo del semestre. El grupo se crea por medio de la URL:

http://maui.ls.fi.upm.es/entrega/CrearGruposProgramacion2.html

El alumno de un grupo que realice la primera entrega de una práctica, será el que tenga que hacer todas las entregas para esa práctica.

Entrega: La práctica se entregará a través de la página web:

http://maui.ls.fi.upm.es/entrega/

El periodo de entrega finaliza el día 11-11 a las 14:00. La práctica entregada debe compilar en la versión 1.6 del J2SE de Oracle/Sun y compatible con JUnit 4.8. En el momento de realizar la entrega, la práctica será sometida a una serie de pruebas que deberá superar para que la entrega sea admitida. El alumno dispondrá de **un número máximo de diez entregas**. Ahora bien, si el alumno realiza **más de cinco** entregas, se le **restará un punto** en la nota final de la práctica. Asimismo, por el hecho de que la práctica sea admitida, eso no implicará que la práctica esté aprobada. **El fichero a entregar será Mecanico.java.**

Código Auxiliar: Para la realización del presente ejercicio se suministra código de apoyo, así como un esqueleto de la clase a implementar dentro del fichero **EjMecanicoCodAlumno.zip** que acompaña a este enunciado. **El código de apoyo y las cabeceras de la clase Mecanico NO han de modificarse**. Como parte del código de apoyo, se proporciona el fichero **reparacion.jar**, que contiene la solución del ejercicio de las Reparaciones (solo el bytecode). Este fichero deberá ser utilizado por todos alumnos tanto si realizaron el ejercicio de las reparaciones como si no.

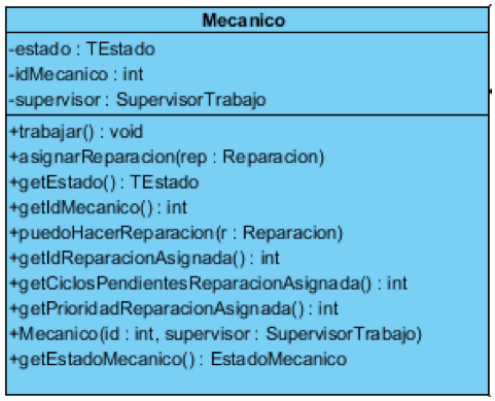
Detección Automática de Copias: Cada práctica entregada se comparará con el resto de prácticas entregadas en todos los grupos de la asignatura. Esto se realizará utilizando un sofisticado programa de detección de copias.

Consecuencias de haber copiado: Todos los alumnos involucrados en una copia, bien por copiar o por ser copiados, quedan inhabilitados para presentarse a todas las convocatorias de examen del presente curso, además de la posible apertura de expediente académico.

Especificación

Un objeto mecánico es un objeto que puede realizar reparaciones normales o prioritarias de una en una. Para que un objeto mecánico pueda realizar una reparación, se le debe asignar dicha reparación. Una vez asignada, el mecánico podrá empezar a trabajar en ella. Para ello, se deberá invocar al método trabajar(), de tal forma que si una reparación requiere N ciclos de trabajo para ser completada, será necesario llamar N veces al método trabajar(). Por otro lado, el objeto mecánico debe notificar los siguientes tres eventos relativos a su actividad a un objeto supervisor:

1. Se ha pospuesto una reparación para realizar otra más prioritaria,
2. Se ha finalizado una reparación y
3. Se ha avanzado un ciclo en la realización de una reparación, pero aún no se ha completado.



Se pide implementar una clase Mecanico que contenga al menos:

Atributos

- **idMecanico**: identificador del mécanico
- **estado**: indica si el mecánico está libre o ocupado.
- **reparacionEnCurso**: es el objeto reparación en el que está trabajando el mecánico.
- **supervisor**: es el objeto al que el mecánico tiene que notificar los tres eventos mencionados anteriormente.

Métodos

- **Constructor**:

PRE: supervisor no nulo
Recibe como parámetros el identificador del mecánico y el supervisor que tiene asignado. Inicializa el objeto con los parámetros y deja al mecánico en estado libre.

- **getCiclosPendientesReparacionAsignada**.

  PRE: Mecánico ocupado

  Retorna los ciclos pendientes de la reparación asignada al mecánico.

- **getEstado**:

  PRE: cierto

  Método que retorna el estado del mecánico.

- **getIdMecanico**:

  PRE: cierto

  Retorna el identificador del mecánico.

- **getIdReparacionAsignada**:

  PRE: Mecánico ocupado.

  Retorna el identificador de la reparación asignada.

- **getPrioridadReparacionAsignada**:

  PRE: Mecánico ocupado.

  Método que retorna la prioridad de la reparación que está realizando. Retorna 0 si la reparación es normal, y en otro caso retorna la prioridad.

- **puedoHacerReparacion**:

  PRE: cierto

  Método que se utiliza para **preguntar** al mecánico si puede atender la reparación que recibe como parámetro. Si puede realizarla retorna cierto, en caso contrario retorna falso. El mecánico podrá atender la reparación no nula que recibe como parámetro si:

  - Está libre,
  - la reparación que está realizando es normal y la que recibe es prioritaria o
  - la reparación que está realizando tiene prioridad p y la que recibe tiene prioridad p' y se cumple p' > p

- **asignarReparacion**:

PRE: El mecánico puede atender esta reparación según lo estipulado en el método anterior.

Se le asigna la reparación al mecánico. En el caso de que el mecánico ya estuviera ocupado con otra reparación, pospone esta reparación notificándoselo al supervisor.

- **trabajar**:

  PRE: El mecánico está ocupado

  Trabaja un ciclo sobre la reparación que tiene asignada y llama al método del supervisor correspondiente para notificarle el estado de progreso de la reparación (véase la descripción del atributo supervisor).

Código de apoyo

Los alumnos deben utilizar el código de apoyo que se encuentra en el archivo **EjMecanicoCodAlumno.zip**. Este archivo contiene:

- JTestMecanico.java. JUnit que sirve para probar la implementación del mecánico. Este JUnit requiere JUnit 4.8.
- reparacion.jar: Librería que tiene la implementación de la jerarquía de reparaciones y todas las excepciones asociadas. Son los binarios de la solución del ejercicio de las reparaciones.
- src/taller: Directorio en el que se encuentra el código del paquete taller que utiliza el JUnit de la clase Mecanico.
- src/mecanico: Directorio en el que se encuentra el código del paquete mecanico, y donde se debe ubicar el fichero Mecanico.java a implementar por el alumno.
- src/mecanico/excepciones: Contiene las implementaciones de las excepciones necesarias.

Se valorará positivamente:

- La utilización de nombres significativos para los identificadores, así como la utilización de los convenios de Sun/Oracle.
- La utilización de métodos auxiliares que implementen tareas comunes a varios métodos, y que de esa forma eviten la duplicidad de código.
- La correcta indentación del código. Se recomienda la utilización en eclipse del atajo de teclado CTRL + i.